UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/735,715	12/12/2000	Jacob Dreyband	033144-004	5590

23562 7590 01/26/2005

BAKER & MCKENZIE
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER	
VU, TUAN A	
ART UNIT	**PAPER NUMBER**
2124	

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/735,715	**Applicant(s)** DREYBAND ET AL.
	Examin r Tuan A Vu	**Art Unit** 2124

-- Th MAILING DATE of this communication appears on the cover sheet with the correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12/30/2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21 and 43-71* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21, 43-71* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 December 2000* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 12/30/2004.

As indicated in Applicant's response, claims 1, 43 have been amended. Claims 1-21, 43-71 are pending in the office action.

Drawings

2. New corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because the legibility of the hand-written legend in the current informal drawing (filed 12/12/2000) creates some difficulties in the visual construction thus semantic interpretation of the invention. Optionally, Applicant is advised to employ the services of a competent patent draftsperson outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-21, and 43-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah, USPN: 6,477,580 (hereinafter Bowman), in view of Francis et al., USPN: 6,665,861 (hereinafter Francis), and further in view of White et al., 6,438,559 (hereinafter White).

As per claim 1, Bowman discloses a method for presenting data within a computing environment including an application program interface (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object for storing data (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187; Fig. 98 – Notes: browser interpreting pages is equivalent to tagged data being stored in message or packages streamed between browser applications or interconnected framework machines);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 98; Fig. 184-185);

packing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191- Note: a stream being passed over the internet reads on binary representation of being packed tagged data being streamed and eventually processed by recipient machine application engines).

But Bowman does not explicitly specify that the package or message of tagged data are storing universal tagged data object being platform independent, hardware independent, and language independent. But in view of Bowman's disclosing of COM format for effecting RPC, messaging utilities and directory services having platform independent standard for transmitting data (e.g. Fig. 20-22; col. 73, lines 10-41; col. 63, line 62 to col. 63, line 21 – Note: COM format data are platform and language neutral by nature of common platform object broker services), in combination of language neutral for Java byte codes (*virtual machines 2706* – Fig. 27), and binary representation across hardware independent internet protocol, the above limitation is at least strongly suggested. The packaging of data using Java platform neutral format was a well-known concept in the art of software transmission at the time the invention was made. Francis, in a method to transmit package of Java binary representation and metadata similar to Bowman streaming of data (Bowman: Fig. 108-109) across computers, also discloses packaging binary representation of Java beans with supporting utilities/metadata under markup or tagged form like XML (Fig. 6-8). In case the platform, hardware and language neutral package stream by Bowman are not universal tagged data for browser use, it would have been obvious for one of ordinary skill in the art at the time the invention was made to encapsulate such data in XML form as taught Francis because this will alleviate resources of the receiving computer in making use of readily formatted data without additional compilation.

Further, Bowman does not explicitly disclose universal tagged data being encapsulated for universal access to manipulation and aggregation of tagged data; however, the very fact of having stream of binary packets across platform discloses encapsulated data for universal access, universal in the context of many machines that can establish reception of such stream; hence Bowman disclose universal tagged data. As for the access to manipulation and aggregation of tagged data, Bowman discloses browser manipulation of tagged data and markup language data manipulation and aggregation using browser application in conjunction with stream, message passing and ORB remote calls using platform independent-based services as mentioned above (e.g. Fig. 13-18; Fig. 98); hence has implicitly disclosed access of tagged (or markup) data for manipulation or aggregation (Note: data provided through COM or ORB services and a compilation of HTML formatted data in pages composed of subdivided markup sections implicitly disclose access for manipulation or aggregation of data).

Nor does Bowman explicitly disclose tagged data object capable of being transferred among and processed by computer environments for processing without any intermediate format conversions. Bowman discloses web format aggregating inherent layers of markup tags, markup data (e.g. Fig. 13-18; Fig. 24, 98), such data being sent turn into a binary stream format for enabling the transmission over different computers through platform independent messaging services or protocols as mentioned above for browser processing without recompilation. This in combination with the rationale as set forth above using Francis' teachings discloses tagged data object (i.e. a binary stream representing a tagged data) being transferred and processed without any intermediate format conversions because browser can make use of markup language as received in browser applications.

But Bowman does not explicitly specify that the encapsulated tagged data provides data that includes data element and a corresponding binary tag id. Bowman or Francis, however, teaches tagged data for interpretation by browsers, hence implicitly discloses a variable name bracketed within the begin tag and end tag; and teaches attributes descriptors in the meta-data section comprising a header section of the object-based stream, which suggests encapsulating identification information of the data element of the stream, according to a well-known concept of including some identifier in a binary form to associate the data bundle or packet sent over a network communication link with its content at the time the invention was made. White, in a method to serialize objects for distribution over a communication network environment using descriptors in serialization of class objects analogous to the object streaming and meta-data by

Bowman or Francis, discloses the tagging of object content being serialized with an identifier or value for packing and deserializing (e.g. *ACI* - col. 4, lines 16-58; col. 10, line 36 to col. 11, line 9). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the tagging technique associating an identifier with the tagged content as taught by White and apply it to the stream metadata by Bowman, in case Bowman's metadata or tagged stream does not include such tag identification already, because this tag ID would facilitate the differentiation between data being packed and enable data handling/re-processing as well as unpacking or modification of elements packed in the message or bundle.

As per claim 2, Bowman discloses the packing of tagged data being a simple object and a complex object, and list object (e.g. col. 124, line 14 to col. 127, line 39 – Note: the use of Java or C++ based components implicitly discloses basic class, compound classes, or structure/enumeration of basic classes and compound classes objects).

As per claim 3, Bowman discloses packing a simple object by retrieving data attributes for length of an object source identifier, object size, type, value; allocating of packed memory location for object identifier length (e.g. col. 235, line 47 to col. 237, line 32); copying the object size, type, and value into the packed memory location (Note: this is inherent to the above cited portions); retrieving and copying head value and exit value into the packed memory location (e.g. *START INDEX, WS-INDEX, STREAM-END* - col. 237, line 35 to col. 238, line 66).

As per claims 4 and 5, Bowman does not explicitly specify the steps of retrieving, writing, and allocating/writing for the complex object as has been disclosed for the simple object from claim 3, but in view of the packaging of data in the retrieval of business-related complex object (e.g. col. 204, line 40 to col. 207, line 59), the limitations as recited are herein implicitly in view of the inherent presence of simple object within complex object or list objects.

As per claim 6 and 7, Bowman does not specify packing list object with retrieving of object source identifier, allocating memory in a packed memory location to accommodate the list object source identifier length; retrieving and copying list head value and list exit value into the packed memory location; but in view of the rationale used in addressing claims 4 and 5, these limitations are also implicitly disclosed because of the inherent presence of simple objects and complex objects in structure or enumeration, i.e. list, object so well-known in object-oriented language.

Further, Bowman does not explicitly disclose retrieving list array object and copying it to the packed memory location. But, in view of the inherent array structure in structure or enumeration of simple and complex objects in C++ or Java, this limitation is also implicitly disclosed as per the same rationale used for claims 4 and 5.

As per claim 8, Bowman does not explicitly disclose that the tagged data object is an universal data container that is platform, language, and architecture independent for access to manipulation and aggregation of structured or unstructured data; but in view of rationale used in claim 1 to address tagged data being hardware, platform and language independent and provided for access to manipulation, aggregation tagged data, the limitation is rejected herein with the same rationale as set forth therein (Note: structure data is formatted data stream according to HTML, XML or internet or COM protocol reads on container; as opposed to unstructured data are generic data used or referenced indirectly by such structured data)

As per claim 9, see Bowman (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191).

As per claim 10, refer to claim 2.

As per claim 11, Bowman discloses data wrapping (e.g. *Wrapper component* - Fig. 81).

As per claim 12, Bowman (col. 131-132; col. 174, line 33 to col. 175, line 24; Fig. 50-51), discloses modeling using COM and Case Tools but Bowman does not specify including of named tree with a field name connected with a value. But in view of the teaching of language-independent modeling along with metadata or language neutral format as addressed in claim 1, (Francis' teaching modeling and tagged web format data is suggesting of tree structure implementation of data to be transmitted as metadata or specification data), this limitation would have obvious for the same rationale as used in claim 1 and also the association of tree with field name as metadata would enhance the utilization and re-processing of data tagged and stored in the package.

As per claim 13, Bowman discloses Java and C++ constructs which inherently include list or enumeration of objects of simple and complex type (see claim 2).

As per claim 14, this claim includes the encapsulation of data type, tag id, and writing thereof to the tagged data object and these limitations have been addressed in claim 3 and 4.

As per claim 15, Bowman discloses the use of Java objects, hence has implicitly disclosed one of the following data type: integer, float, byte, char string, a java object, a null data, a primitive type, a compound type, and a list type.

As per claim 16, Bowman (in combination with White/Francis) discloses tag identifier with of type integer (see White from claim 1).

As per claim 17, Bowman with White's teachings discloses serializing of tagged data and compacting it in a stream for transmission, hence has implicitly disclosed a tagging process following a linear sequence, i.e. sequential tagging with determining a sequence.

As per claim 18, Bowman with White's teachings discloses including a data, a position and a tag element (refer to claim 3; Fig. 109 – Note: Index position use in writing data by Bowman discloses including an position and packet layout inherently encompasses boundaries position of data compacted in packet).

As per claim 19, Bowman does not explicitly specify converting of first type of tagged data to second type of data for a change in properties; but the concept for converting the order of data type (e.g. network-bound integer converted into local host-based integer and vice-versa, as per Java/C++ *ntohs* or *htons* functions) for allowing data type to be communicated through the internet medium was a well-known concept at the time the invention was made. Hence, Bowman's disclosed communication of Java or C++ objects implicitly discloses such conversion to provide for a communication properties adjustment or change as claimed.

As per claims 20 and 21, by virtue of the rejections of claim 2 and claim 19 above, the limitations of these claims are implicitly disclosed.

As per claim 43, Bowman discloses a method for presenting data within a computing environment including an application program interface prescribed for data conversion and wire formatting specification (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187); wherein the tagged data object comprises a universal data container that is platform and hardware independent (e.g. col. 99, lines 7-40 – Note: Java platform independency is implicitly disclosed); said tagged data object providing broad access to manipulation and aggregation of structured data and unstructured data (e.g. *view configurator, maximum maintainability and extensibility* - col. 248, line 28 to col. 259, line 45; *LUW* - Fig. 108-129, 163-191 – Note: context retrieving and selecting appropriate objects from requests is equivalent to broad access for data manipulation and aggregation);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 184-185);

providing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191);

transmitting the tagged data transmission (e.g. Fig. 105-107);

unpacking the tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29).

But Bowman does not explicitly disclose tagged data object to provide universal access to manipulation and aggregation of a structured data and unstructured data; but this limitation has been addressed in claim 1 and 8 above.

Nor does Bowman explicitly disclose that the packed tagged data object is capable of being transferred among and processed by computer environments without any intermediate format conversions. As mentioned above, Bowman discloses web format aggregating inherent layers of markup tags, markup data (e.g. Fig. 13-18; Fig. 98), such data being sent turn into a binary stream format for enabling the transmission over different computers through platform independent messaging services or protocols as mentioned above for browser processing without recompilation. This in combination with the rationale as set forth above using Francis' teachings discloses tagged data object (i.e. a binary stream representing a tagged data) being transferred and processed without any intermediate format conversions because browser can make use of markup language as received in browser applications.

Nor does Bowman explicitly specify that the encapsulated tagged data object includes a data element and a corresponding tag id; but this also has been addressed in claim 1 above using Francis/White.

As per claims 44-49, these claims correspond to claims 2-7 respectively; hence are rejected likewise, respectively.

As per claim 50, this corresponds to claim 2, and is rejected using the rationale of claim 2. Further, Bowman discloses a method for presenting data within a computing environment including an application program interface comprising the steps of unpacking tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29-- Note: in view of the teachings on packing data into package or stream to be sent in packet over the internet by Bowman as mentioned in claim 1, the steps of unpacking, creating a storage for the unpacked data received over the internet, and the extracting of object being tagged are implicitly disclosed).

As per claim 51, Bowman does not specify the steps of retrieving the simple head value and simple exit value; allocating memory in an unpacked memory and copying of simple object size, type and value into said unpacked memory. Official notice is taken that subjecting packets received from the internet into a host or routing, or a gateway machines to unpacking and buffer storage was a well-known concept in the art at the time the invention was made. In view of the teachings for unpackaging of data by Bowman above and the well-known unpacking of data, it would have been obvious for one skill in the art at the time the invention was made to provide the unpacking of the tagged data as taught by Bowman/Francis/White using the well-known technique of unpacking/storage above because this would enable correct extraction of data based on boundaries locations and allocation of correct memory resources.

As per claims 52 and 53, the limitations as to unpack a complex object would also have been obvious by virtue of the inherency of simple object in a complex objects as mentioned in claims 4-5 and the rejection used in claim 51 above.

As per claims 54 and 55, the rationale used for claims 6-7 and 52-53 are herein applied.

As per claims 56-59, refer to rejections of claims 10-13 respectively.

As per claim 60, this claim corresponds to claim 14, hence is rejected using the same rationale as set forth therein.

As per claims 61-67, refer to corresponding rejections of claims 15-21 respectively.

As per claim 68, Bowman does not specify extracting data with determining the type to provide the tag id; and writing the data element into the tagged data object. But in view of White's or Francis's teaching to provide a tagging associated with an identifier in order to facilitate the reprocessing of data manipulated at the receiving end and the rationale for encapsulating in claim 14, this step would have been obvious because the implied and inherent association between packing and unpacking.

As per claims 69 and 70, see rejection of claims 15 and 16 respectively.

As per claim 71, in view of the unpacking as taught by Bowman and the rationale in claim 18 above, this limitation would also have been obvious by virtue of the adding of element in the tagged data as mentioned in the above rejection.

Response to Arguments

5. Applicant's arguments with respect to mainly claims 1, 43 have been considered but are not persuasive. Following are the Examiner's corresponding counter arguments and observations thereto.

(A) Applicants have submitted that 'browser applications do not and cannot exchange data objects of the type recited … which are universal and thus platform, architecture … independent'; and that 'browsers employ HTML and XML formats to process[es] data they receive … HTML and XML data as serialized data transmission in a textual format. Then, … HTML, XML, etc., is converted to a binary… Once any processing is completed … back to a textual' (Appl. Rmrks, pg. 15, bottom, pg. 16, 1st and 2nd para). The claims recites 'a binary representation of the tagged data object capable of being transferred among and processed by … without any intermediate format conversions'. Bowman discloses a stream object including tagged data; and that reads on binary representation because in the world of transmission across the internet, there is only stream of 1's and 0's at the physical level, among other layers of the OSI network hierarchy; and not until the application level, do these binary forms or objects become application-specific formats, among which graphical display or textual. As for the 'object capable of (emphasis added) being transferred and processed without any intermediate format conversions' limitation, the claim recites 'capable of'. Such limitation does not enforce any action that actually generates some patentable results, whether or not it is a transmission result or a conversion result being effectuated at the receiving environment. Granted that such object can potentially be (as in capable of) subject to transformation or processing by an computer environment; the word 'environment' is not specific enough to preclude a lot of interpretations. One such interpretation is that Bowman's stream of data gets into the low-level socket port of the receiving machine (and this is inherent feature), and that the unpacking performed therein reads on the processing by an environment of such binary object. It also reads on direct processing without any intermediate conversion. Another interpretation is that the packet of binary data gets manipulated or rerouted to another environments as shown in Bowman's Fig. 18, 24. The claim is not showing metes and bounds in regard to how the processing by the 'computer environments' is clearly about. Nor does the claim preclude the eventuality that the computer after processing the incoming data, channels it to the application layer for conversion into a human readable form. As implied from the arguments, the processing incurs in the context of a browser parsing XML language. As interpreted from the claimed language, that is far from the case. The recited 'processed by' does not start and stop a just converting data at a browser level, nor does it enforce only a browser receiving and processing while excluding any form of low level processing by the physical layer. As interpreted, the limitation 'processed by computer environments without any intermediate form conversions' is not specific with respect to at least three main concepts. First, the 'computer environments' does not enforce an browser application as asserted by Applicants; second, the 'processed by ... without any intermediate format conversions' does not necessarily signify 'computer application strictly and directly using a binary formatted stream and not converting a textual stream into a series of 1s and 0s; third, the 'without any intermediate ... conversion' phrase does not enforce conversion from text to binary or vice versa, notably for the simple reason that claiming a limitation with 'without...' is not providing scope and limits on how something should happen or not happen. As shown in the rejection, Bowman discloses receiving packet of streamed data formatted in a special binary form and process the tagged data included therein. The process of making use (at the implied lower level of OSI hierarchy) of such streamed data as they arrive is considered a direct process without any intermediate form. Maybe, later the application layer, another processing involves conversion; but the claim does not set clear bounds as to what constitutes a 'processed by' limitation; what constitutes a 'computer environments' or under what time frame is this 'processed ... without any intermediate ... conversions' happening; nor is the claim clear on what is required under this ' without any intermediate format conversions' ambit. For the sake of argument, even if the fact of processing by the browser application includes a conversion of tagged data into other format, such conversion is not being undeniably precluded by the way the claim is explicitly put together. This is simply because processing by a computer environment without any intermediate conversion is broad in 3-4 counts: *capable of, processing, environments, without an intermediate format conversions*; and as explained above, the teaching by Bowman, explicitly or implied, has fulfilled those limitations as they are formulated in the claim.

(B) The Applicants have submitted that as a result of text-based format conversion, Bowman does not disclose or even suggest 'where the tagged data object is represented in a universal binary format ... format' (Appl. Rmrks, pg. 16, bottom, pg. 17, 1st para). Bowman discloses a stream that can go across platform regardless of the nature of data language included or the language implementing the receiving applications. And that reads on the universal binary format; because the term 'universal' is broadly recited and does not entail more than the implied and previously recited platform, architecture, language independency. Thus, Bowman's binary stream fulfills this limitation. As to the textual conversion, this assertion has been addressed in section A above.

(C) The Applicants have submitted that concerning the limitation 'tagged data object that is ... language independent', Bowman's COM as well as Java bytecodes do not suggest the type of universal independence as claimed; and that ORB services require a broker and is not based on independence of data format (Appl. Rmrks, pg. 17). The rejection has pointed out that the use of COM, ORB, or Java are there to address the universal access to manipulation and aggregation of tagged data. By virtue of the stream data being formatted to reach the client ports and server ports as taught by Bowman, this access is universal in the context that such data is received by every machine for processing. In the other hand, the binary formatted nature of the stream from Bowman as mentioned in section A above is used to address the universal aspect of tagged data object. Therefore, the arguments that Java or COM or ORB are not fulfilling the universal independence are moot, misplaced or based on a limitation that does not seem to come from the claim (i.e. the claim does not recite 'universal independence') because the limitation universal tagged data object being platform, architecture, language independent means just what they are literally recited; and cannot be equal to 'universal independence', which is also a very broad term and otherwise would require explicit specificity and further descriptive limitations. In other words, the rejection has pointed out what in Bowman reads on universal tagged data; and what reads on universal access to manipulation and aggregation. Besides, the claim does not define the so-called 'universal access to manipulation and aggregation' in terms as to indisputably preclude what in Bowman's has been cited to meet such limitation. Thus, the arguments amount to mere assertions without setting forth what in the invention clearly distinguishes over the cited parts of a reference, e.g. Bowman's.

(D) Applicants have submitted that Francis's metadata and the XML textual format do not amount to an universal accepted language; and that XML require serialized data format for transmission; and all does not meet the transmission of a machine-level binary representation of the tagged data as claimed (Appl. Rmrks, pg. 18; pg. 19, top). First, the textual aspect of XML being converted into a binary form fall under the ambit of the response set forth in section A; thus is referred back there, because based on the claim interpretation, there is no explicit specifics in the claims to preclude that data being streamed over to different machines does not meet the universal tagged data object transferred and processed without intermediate conversion. Nor does the claim specify that the binary format is strictly format such as a compiled machine executable form for example, or a machine-level code as asserted by Applicants. A binary form representation has been set forth in the rejection as internet-based binary data being streamed over and included tagged data; and Bowman has met such binary representation as mentioned in section A. The claim does not laid out specifics as to how the tagged data is incorporated into the binary representation so as to overcome Bowman's stream of binary object containing markup tags data. As mentioned in section A, the claim does not set metes and bounds for the limitation 'capable of ... processed by ... environments without any intermediate format conversion' so that it would clearly preclude the possibility that the tagged data is a textual stream being converted into a serialized form or vice versa. Besides, even if Francis teaches the transmission of textual files, the transmitted stream is a binary form and received for processing at the receiving port. As mentioned in section A, such paradigm reads on the claim limitation as recited because in part to the relative broad terminology used in the claim.

(E) Applicants have submitted that White does disclose tagged data in a textual format and does not cure the deficiencies of Bowman and Francis (Appl. Rmrks, pg. 19, middle para). The rejection using White is for addressing the limitation of tag ID; not for covering deficiencies coming from converting textual data as asserted by Applicants. It is apparent that the arguments are not directed to the very rationale combining 3 references to address the missing teaching from one of the references (e.g. Bowman) used; but instead contends with exposing a deficiency related to one limitation in one reference (e.g. White) which has already been met from other references (e.g. Bowman or Francis). In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Section A has set forth the reasons why Applicants' arguments based on from the XML intermediate conversion assertion is not persuasive; nor does such alleged conversion be enforced as one and only interpretation in view of the way the claimed limitation is recited. Hence, Applicants fail to specifically point out why the rationale to combine Bowman, Francis and White would generate adverse effects or stem from improperly combined methodologies.

(F) Concerning Applicants request that the 103(a) rejection be withdrawn (Appl. Rmrks, pg. 20), this is the response. Most of the arguments evolve around the universal aspect of the tagged data object; and set out behind the rationale of improved processing time (which is not claimed) wherein, as asserted by Applicants, no textual data conversion is needed when compared to Bowman's metadata, XML, COM or ORB-based known methodologies. Yet the claims seem to lack limitations that would establish continuity or causality between the recited universal tagged data limitation and the benefits implied from the arguments as to why such tagged data when processed necessitate no additional conversion, i.e. some crucial functional or structural steps or limitations that would prevent the packed data from being additionally transformed before its tagged content is used by an application level. The rejection only address the claims in view of wherever broad and reasonable interpretation permits; and while Applicants identify what can be construed as teaching away from the invention, Applicants have failed to see why the rationale as set forth in the rejection has only addressed the limitations in light of what is recited and thus interpreted; with such interpretation being the results from the broadness of the claim and the void between elements claimed as mentioned above. For this reasons, the rejection stand rejected as set forth above.

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
January 19, 2005

ANIL KHATRI
PRIMARY EXAMINER